

IN THE CLAIMS:

Please cancel claim 37.

Please amend the claims as follows:

1 1. (CURRENTLY AMENDED) Apparatus for enabling an instruction to control data
2 flow bypassing hardware within a processor of a programmable processing engine, the
3 apparatus comprising:

4 a pipeline of the processor, the pipeline having a plurality of stages including
5 instruction decode, writeback, and execution stages, the execution stage having a
6 plurality of parallel execution units; and

7 an instruction set of the processor, the instruction set defining a first register
8 decode value, ~~that specifies one of a first register decode value which~~ that defines
9 source operand bypassing that allows source operand data to be shared among the
10 plurality of execution units, and a second register decode value that defines result
11 bypassing that allows bypassing of a result from a previous instruction executing in
12 pipeline stages of the processor.

1 2. (ORIGINAL) The apparatus of Claim 1 further comprising:

2 a register file containing a plurality of general-purpose registers for storing
3 intermediate result data processed by the execution units; and

4 a memory for storing one of transient data unique to a specific process and
5 pointers referencing data structures.

1 3. (CURRENTLY AMENDED) The apparatus of Claim 1 wherein the second register
2 decode value comprises:

3 one of a result bypass (RRB) operand and an inter-unit result bypass (RIRB)
4 operand, each of which explicitly controls data flow within the pipeline of the processor.

FI
ET

1 4. (ORIGINAL) The apparatus of Claim 3 wherein the execution units comprise a
2 current execution unit and an alternate execution unit, and wherein the RRB operand
3 denotes the current execution unit and the RRB operand denotes the alternate execution
4 unit.

1 5. (ORIGINAL) The apparatus of Claim 3 wherein the RRB operand explicitly infers
2 feedback of the data delivered from a current one of the execution units to an input
3 register of the current execution unit over a feedback path.

1 6. (ORIGINAL) The apparatus of Claim 5 wherein the writeback stage comprises an
2 interstage register and wherein the RRB operand enables bypassing write-back of the data
3 processed by the execution units to one of the register file or the interstage register of the
4 writeback stage.

1 7. (CURRENTLY AMENDED) The apparatus of Claim 2 wherein the first register
2 decode value comprises a source bypass (RISB) operand that allows source operand data
3 to be shared among the parallel execution units of the pipelined processor.

1 8. (ORIGINAL) The apparatus of Claim 7 wherein the execution units comprise a main
2 execution unit and a secondary execution unit, and wherein the RISB operand allows the
3 secondary execution unit to receive data stored at an effective memory address specified
4 by a displacement operand in the previous instruction executed by the main execution
5 unit.

1 9. (CURRENTLY AMENDED) A method for enabling an instruction to control data
2 flow bypassing hardware within a pipelined processor of a programmable processing
3 engine, the method comprising the steps of:

4 defining a first register decode value that ~~specifies one of a first register decode~~
5 ~~value which defines source operand bypassing~~ of source operand data and a second

6 register decode value that defines result bypassing of a result from a previous instruction
7 executing in pipeline stages of the processor; and
8 identifying a pipeline stage register for use as a source operand in an instruction
9 containing the first or the second register decode value.

1 10. (CURRENTLY AMENDED) The method of Claim 9 further comprising: ~~the step of~~
2 explicitly controlling data flow within the pipeline stages of the processor through
3 the use of a register result bypass (RRB) operand in said second register decode value.

1 11. (CURRENTLY PRESENTED) The method of Claim 10 further comprising:
2 including pipeline stages having instruction decode, writeback and execution
3 stages, and wherein the execution stage has multiple parallel execution units including a
4 current execution unit and an alternate execution unit.

1 12. (CURRENTLY AMENDED) The method of Claim 11 wherein the step of
2 explicitly controlling comprises ~~the steps of~~:
3 retrieving data from the current execution unit; and
4 returning the data to an input execution register specified by the RRB operand,
5 thereby bypassing write-back of the data to either a register file or memory at the
6 writeback stage.

1 13. (CURRENTLY AMENDED) The method of Claim 12 wherein the step of
2 identifying further comprises ~~the steps of~~:
3 explicitly specifying the pipeline stage register to be used as the source operand
4 for the instruction.

1 14. (PREVIOUSLY PRESENTED) The method of Claim 13 further comprising:
2 encoding the RRB operand in fewer bits than a regular register operand.

1 15. (PREVIOUSLY PRESENTED) The method of Claim 9 further comprising:

2 including pipeline stages having instruction decode, writeback and execution
 3 stages, and wherein the execution stage has multiple parallel execution units including a
 4 current execution unit and an alternate unit; and

5 sharing source operand data among the parallel execution units of the pipelined
 6 processor through the use of a source bypass (RISB) operand in said first register decode
 7 value.

1 16. (CURRENTLY AMENEDDED) The method of Claim 15 wherein the step of sharing
 2 further comprises: the step of

3 receiving data at the alternate execution unit, the data stored at a memory address
 4 specified by a displacement operand in a previous instruction executed by the current
 5 execution unit of the processor.

1 17. (CURRENTLY AMENEDDED) The method of Claim 16 wherein the step of sharing
 2 further comprises: the step of

3 realizing two memory references through the use of a single bus operation over a
 4 local bus.

1 18. (CURRENTLY AMENEDDED) The method of Claim 17 wherein the step of sharing
 2 further comprises: the step of

3 encoding the RISB operand with substantially fewer bits than those needed for a
 4 displacement address.

1 19. (PREVIOUSLY PRESENTED) A computer readable medium containing executable
 2 program instructions for enabling an instruction to control data flow bypassing hardware
 3 within a pipelined processor of a programmable processing engine, the executable
 4 program instructions comprising program instructions for:

5 defining a first register decode value that specifies one of a first register decode
 6 value that defines source operand bypassing of source operand data and a second register

EX
FI

7 decode value that defines result bypassing of a result from a previous instruction
8 executing in pipeline stages of the processor; and
9 identifying a pipeline stage register for use as a source operand in a current
10 instruction containing the register decode value.

1 20. (ORIGINAL) The computer readable medium of Claim 19 further comprising program
2 instructions for explicitly controlling data flow within the pipeline stages of the processor
3 through use of a register result bypass operand.

1 21. (ORIGINAL) The computer readable medium of Claim 20 further comprising
2 program instructions for sharing source operand data among parallel execution units of
3 the pipelined processor through the use of a source bypass operand.

1 22. (CANCELLED)

1 23. (CANCELLED)

1 24. (CANCELLED)

1 25. (CANCELLED)

1 26. (CANCELLED)

1 27. (CANCELLED)

1 28. (PREVIOUSLY PRESENTED) A processor comprising:
2 a first execution unit having at least one first input and a first output;
3 at least one second execution unit having at least one second input and a second
4 output;

FI 5 a first input register connected to said at least one first input;
6 a second input register;
7 a multiplexer having a first input from said first input register, a second input
8 from said second input register, and an output to said at least one second execution unit;
9 and
10 a register decode value that specifies bypassing data from said first input register
11 to said at least one second execution unit via said multiplexer.

1 29. (PREVIOUSLY PRESENTED) The processor of claim 28 further comprising:

2 a first instruction having at least one first source operand and a first destination,
3 said first execution unit processing said first instruction;

4 a second instruction having at least one second source operand and a second
5 destination operand, said at least one second source operand is the same as said at least
6 one first source operand; and

7 means for replacing said at least one second source operand with said register
8 decode value.

1 30. (PREVIOUSLY PRESENTED) The processor of claim 29 further comprising:

2 a register file connected to said first input register and said second input register;
3 and

4 means for loading said at least one first and said at least one second source
5 operands from said register file.

1 31. (PREVIOUSLY PRESENTED) The processor of claim 29 further comprising:

2 a memory connected to said first input register; and

3 means for loading said at least one first and said at least one second source
4 operands from said memory.

1 32. (PREVIOUSLY PRESENTED) The processor of claim 29, said means for replacing
2 further comprising:

3 an instruction decode mechanism; and

4 means for said multiplexer choosing input from said first input register.

1 33. (PREVIOUSLY PRESENTED) The processor of claim 29 further comprising:

2 said register decode value having fewer bits than said at least one second source
3 operand.

1 34. (PREVIOUSLY PRESENTED) The processor of claim 29 further comprising:

2 a displacement value within said at least one first and said at least one second
3 source operands, said displacement value specifying an effective memory address where
4 data is stored.

1 35. (PREVIOUSLY PRESENTED) The processor of claim 29 further comprising:

2 a displacement value within said first destination operand, said displacement
3 value specifying an effective memory address where data is stored.

1 36. (PREVIOUSLY PRESENTED) Electromagnetic signals propagating over a computer
2 network comprising:

3 said electromagnetic signals carrying instruction for execution on a processor for
4 performing the method of claim 9.

1 37. (CANCELLED)

1 38. (PREVIOUSLY PRESENTED) The method of Claim 9 further comprising:

2 including pipeline stages having instruction decode, writeback and execution
3 stages, and wherein the execution stage has multiple parallel execution units including a
4 current execution unit and an alternate execution unit; and

5 explicitly controlling data flow within the pipeline stages of the processor through
6 the use of a register result bypass (RIRB) operand to bypass the writeback stage and to

7 allow result data from an alternate execution unit to flow directly to an input execution
8 register.

1 39. (PREVIOUSLY PRESENTED) The apparatus of Claim 3 wherein the RIRB operand
2 explicitly infers feedback of the data delivered from an alternate one of the execution
3 units to an input register of the current execution unit over a feedback path.

1 40. (PREVIOUSLY PRESENTED) Apparatus for enabling an instruction to control data
2 flow within a processor of a programmable processing engine, the apparatus comprising:
3 a pipeline of the processor, the pipeline having a plurality of stages including
4 instruction decode, writeback and execution stages, the execution stage having a plurality
5 of parallel execution units;
6 a multiplexer connecting parallel execution units; and
7 an instruction set of the processor, the instruction set defining a register decode
8 value that controls said multiplexer to bypass a source operand from a previous
9 instruction executing in pipeline stages of the processor to the source operand of a current
10 instruction.

1 41. (PREVIOUSLY PRESENTED) The apparatus of Claim 40 further comprising:
2 a register file containing a plurality of general-purpose registers for storing
3 intermediate result data processed by the execution units.

1 42. (PREVIOUSLY PRESENTED) The apparatus of Claim 40 further comprising:
2 a memory for storing one of transient data unique to a specific process and
3 pointers referencing data structures.

1 43. (PREVIOUSLY PRESENTED) The apparatus of Claim 40 wherein the register
2 decode value comprises:
3 a source bypass operand (RISB) that allows source operand data to be shared
4 among the parallel execution units of the pipelined processor.

ET
FI

1 44. (PREVIOUSLY PRESENTED) The apparatus of Claim 40 wherein the execution
2 units comprise:
3 a main execution unit and a secondary execution unit, wherein the RISB operand
4 allows the second execution unit to receive data stored at a memory address specified by
5 a displacement operand in the previous instruction executed by the main execution unit.

1 45. (PREVIOUSLY PRESENTED) The apparatus of Claim 44 wherein the instruction
2 set of the processor comprises:
3 an opcode directed to the main execution unit, said opcode having sufficient bits
4 to encode a displacement operand;
5 an opcode directed to the secondary execution unit; and
6 micro-opcodes to initiate memory prefetches without requiring a dedicated
7 instruction.

1 46. (PREVIOUSLY PRESENTED) A method for enabling an instruction to control data
2 flow within a pipelined processor of a programmable processing engine, the method
3 comprising the steps of:
4 defining a register decode value that specifies one of source operand bypassing
5 from a previous instruction executing in pipeline stages of the processor; and
6 identifying a pipeline stage register for use as a source operand in an instruction
7 containing the register decode value.

1 47. (PREVIOUSLY PRESENTED) The method of Claim 46 further comprising:
2 including pipeline stages having instruction decode, writeback and execution
3 stages, and wherein the execution stage has multiple parallel execution units including a
4 current execution unit and an alternate execution unit.

1 48. (PREVIOUSLY PRESENTED) The method of claim 47 further comprising:

2 sharing source operand data among the parallel execution units of the pipelined
 3 processor through the use of a source bypass (RISB) operand.

1 49. (PREVIOUSLY PRESENTED) The method of claim 48 further comprising:
 2 receiving data at said alternate execution unit, the data stored at a memory address
 3 specified by a displacement operand in a previous instruction executed by said current
 4 execution unit of the processor.

1 50. (PREVIOUSLY PRESENTED) The method of claim 49 further comprising:
 2 realizing two memory references through the use of a single bus operation over a
 3 local bus.

1 51. (PREVIOUSLY PRESENTED) The method of claim 49 further comprising:
 2 encoding the RISB operand with substantially fewer bits than those needed for a
 3 displacement address.

1 52. (CURRENTLY AMENDED) Electromagnetic signals propagating on a computer
 2 network, comprising:
 3 said electromagnetic signals carrying instruction for the practice of the method of
 4 ~~Claim 9 or Claim 46.~~

1 53. (CURRENTLY AMENDED) A computer readable media comprising:
 2 said computer readable media containing executable program instruction for the
 3 practice of the method of ~~Claim 9 or Claim 46.~~